

i

Eksamensinformasjon

Eksamensinformasjon – digital skoleeksamen

Fakultet: Teknologi, kunst og design

Utdanning: Teknologiske fag

Emnenavn: Algoritmer og Datastrukturer

Emnekode: (ORD) DATS2300 / ITPE2300

Dato: 19.12.2017

Tid: 15.00 - 18.00

Antall oppgaver: 6

Tillatte hjelpemidler: Ingen

Merknad:

Råd og tips: Bruk ikke for lang tid på et punkt. Gå isteden videre til neste punkt og eventuelt tilbake hvis du får god tid. Resultatet fra et punkt som du ikke har løst, kan brukes senere i en oppgave som om det var løst. Prøv alle punktene. De 6 deloppgavene teller likt, Hvis du skulle trenge en hjelpestruktur (liste, stakk, kø o.l.) fra java.util eller fra kompendiet, kan du fritt bruke den uten å måtte kode den selv. Men den må brukes på en korrekt måte. Men du bør si fra om dette i en kommentar. Hvis du har idéer om hvordan ting skal løses, men likevel ikke klarer å få det til, kan du demonstrere idéene dine med ord, tegninger o.l.

1 **Oppgave 1 / Oppgåve 1**

Lag metoden **public static int *frekvensFordeling*(int[] a)**. *a* skal være sortert stigende, og hvis *a* ikke er det, skal det kastes en `IllegalStateException` (med en passende tekst). Metoden skal returnere de forskjellige verdiene i *a* og hvor mange ganger de dukker opp. Hvis f.eks. *a* inneholder 3, 3, 4, 5, 5, 6, 7, 7, 7 og 8, skal metoden returnere

3 2
4 1
5 2
6 1
7 3
8 1

Metoden skal **ikke** endre noe på tabellens innhold. Hvis tabellen er tom (har lengde 0), skal metoden returnere 0 0

Med andre ord er ikke en tom tabell en feilsituasjon.

Skriv ditt svar her...

1	
---	--

Maks poeng: 10

2

Oppgave 2/Oppgave 2

En datastruktur *s* av typen *Iterable* (se vedlegget) har garantert en iterator. Ved hjelp av den kan datastrukturen traverseres. Lag den generiske metoden

public static <T> T maks(Iterable<T> s, Comparator<? super T> c).

Den skal returnere største verdi i *s*. Sammenligningene gjøres ved hjelp av komparatoren *c*. Hvis *s* er tom (det kan avgjøres ved hjelp av *iterator*en), skal det kastes et unntak.

Vedlegg:

```
public interface Iterable<T>
{
    Iterator<T> iterator();
}
```

Skriv ditt svar her...

1

Maks poeng: 10

³ Oppgave 3/Oppgave 3

En melding inneholder kun tegnene A, B, C, D, E, F, G og H. Tegnenes frekvensfordeling gir følgende Huffmante der hvert tegn står under tilhørende bladnode (se vedlagt pdf-dokument)

I et binærtre har hver node en posisjon. Rotnoden har posisjon 1. Hvis en node har posisjon k , har venstre barn posisjon $2k$ og høyre barn posisjon $2k + 1$.

1. Sett opp posisjonene til bladnodene i Huffmantreet.
2. Sett opp de forskjellige bitkodene som treet gir for de åtte tegnene.
3. En melding som ble komprimert ved hjelp av disse bitkodene, ble komprimert til 00001101111010100111. Dekomprimer dette.

Skriv ditt svar her...

Maks poeng: 10

4 Oppgave 4/Oppgave 4

I denne oppgaven skal du tegne to binære søketrær.

Gitt tallene 12, 7, 11, 3, 17, 15, 6, 8, 10, 4, 16, 13, 14, 5. Legg dem inn, i den gitte rekkefølgen, i et på forhånd tomt binært søketre.

Tegn treet! Hvilken høyde har treet?

Du skal nå legge inn de samme tallene i et binært rød-svart søketre, marker røde noder som firkanter, og sorte noder som rundinger.

Tegn treet etter at du har lagt inn 12, 7, og 11. Hvilken høyde har dette treet?

Tegn treet etter at du i tillegg har lagt inn 3, 17, og 15. Hvilken høyde har dette treet?

Tegn det ferdige treet (etter at alle tallene er lagt inn). Hvilken høyde har dette treet?

Skriv ditt svar her...

Maks poeng: 10

5 **Oppgave 5/Oppgåve 5**

Klassen **LenketHashTabell** bruker "lukket adressering med separat lenking". Den inneholder en *tabell* med *nodereferanser* der alle i utgangspunktet er null. Et *objekt* legges inn på objektets *tabellindeks* (objektets hashverdi modulo tabellengden). Dvs. en *node* (med objektet) legges først i den (eventuelt tomme) lenkede nodelisten som hører til tabellindeksen.

En samling navn (tegnstrenger) skal legges inn. Det er en jobb å regne ut hashverdier og indekser for hånd. Dette er derfor allerede gjort for noen lengder/dimensjoner:

navn	Espen	Bo	Ali	Petter	Karl	Siri	Muhammad	Mari	August	Åse
hashverdi	80088	2357	65968	89125	69562	257197	7934	23763	65085	1983
hashverdi % 13	8	4	6	10	12	5	4	12	7	7
hashverdi % 17	1	11	8	11	15	4	12	14	9	11
hashverdi % 19	3	1	0	15	3	13	11	13	10	7

Setningen `LenketHashTabell<String> hash = new LenketHashTabell<>(n);` oppretter en instans av klassen der den interne tabellen får dimensjon (lengde) lik **n**. Legg inn én og én verdi (*leggInn*-metoden med et navn som verdi) i den gitte rekkefølgen (dvs. Espen, Bo, Ali, osv). En node skal ha både *verdi* og *hashverdi*, men på en tegning holder det med *verdi*.

Utfør følgende oppgave med en lenket hashtabell av lengde 13

- Lag en tegning av datastrukturen når de åtte første navnene er lagt
- Lag så en tegning som viser når alle navnene er lagt inn

Skriv ditt svar her...

Format

B

I

U

x_2

x^2

I_x

ABC

Words: 0

Maks poeng: 10

6

Oppgave 6/Oppgåve 6

En TabellKø oppfører seg som en vanlig kø. Dvs. *leggInnBak()* legger en verdi bakerst i køen, *taUtForan()* tar ut den første verdien i køen, og tilsvarende for *leggInnForan()* og *taUtBak()*. Hva blir utskriften i følgende programbit:

```
Kø<Character> kø = new TabellKø<>();
char[] c1 = "ABCDEFGHI".toCharArray(), c2 = "JKLM".toCharArray();

for (char bokstav : c1) kø.leggInnForan(bokstav);
for (int i = 0; i < 8; i++) kø.taUtBak();
for (char bokstav : c2) kø.leggInnBak(bokstav);
System.out.println(kø); // skriver ut køen
```



Klassen TabellKø bruker internt en såkalt sirkulær tabell. Se figuren i vedlagt pdf-dokument. Der refererer indeks *fra* til den første i køen og *til* til den første ledige plassen (en bak den siste). Legg inn N, O, P og Q i den sirkulære køen med *leggInnForan()* og ta så ut seks verdier med *taUtBak()*. Tegn så køen (du behøver ikke bruke grå bakgrunn)! Hvor mange verdier har køen?




Skriv ditt svar her...



Format



-



B*I*U \times_2 \times^2 \mathcal{I}_x















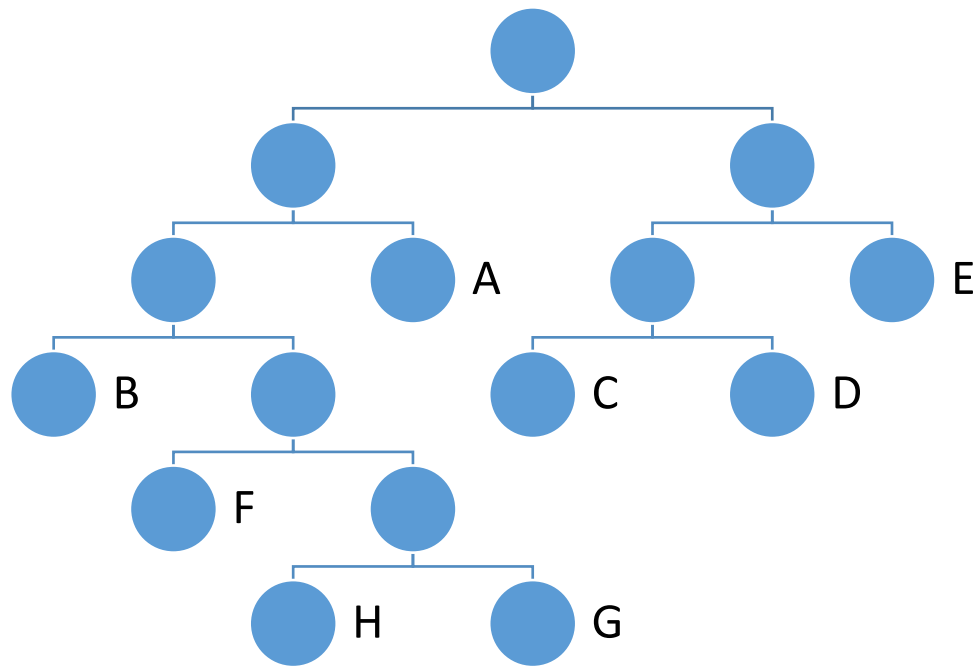
Words: 0

Maks poeng: 10

Question 3

Attached





Question 6

Attached



